VZ LINK magazine

ISSUE 46 AUGUST 1990

New Zealand VZ 300/200 Computer users club

FROM ME TO YOU

Welcome to another issue of your magazine VZ LINK. Hopefully there is something for everyone this issue.

THIS ISSUE

We have yet another superb program written by Ken Clarke. This program called "Resistor Finder" will help you memorize resistor colour codes.

Also we continue the Micro Magic Workshop series. This series is now drawing to an end. I hope we have all managed to get something from the various subjects that have been covered. Gordon Browell did a really fine job putting it together, don't you think ?.

We reprint a nice little program called TeaPot Ted which appeared in the Bumper Book of games put out by ' your computer magazine.
Teapot Ted is also reviewed by Brenda. I have also included a preview of the PCB MINICAD DESIGNER program.

Last but not least we have a very useful utility program which will enable you to 'RESTORE' programs which you NEW in error.

THE COVER

This picture, which goes with the weather we are having at the moment is redrawn from IBM by myself using Art Gallery written by Bruce Kitch.

LIBRARY

I am organising a library of books and magazines for club members to borrow, which will be done on a donation basis.
So if you have any magazines or books that you have finished with (not necessarily connected with the VZ) then please bring them to the next meeting or deliver them to me.

HIGH SCORES

There has not been much input for the High Scores. I have only had two people submit scores do far.
How about the rest of you game players out there, can't stand a little bit of competition,eh.

P.O BOX 1972 C.P.O AUCKLAND N.Z

(C) AUCKLAND VZ 300/200 COMPUTER USERS CLUB (1986)
NEW ZEALAND VZ 300/200 COMPUTER USERS CLUB (1989)

ALL MATERIAL IN THIS NEWSLETTER IS SUBJECT TO COPYRIGHT. CONTRIBUTED PROGRAMS & ARTICLES CAN BE REPRODUCED ON THE UNDERSTANDING THAT IT IS FOR THE PRIVATE USE OF SUBSCRIBERS ONLY.
COPYRIGHT IS ALWAY'S RETAINED BY THE AUTHOR.
THIS CLUB WILL NOT BE HELD RESPONSIBLE FOR ANY BREECHING OF COPYRIGHT CAUSED BY THE PUBLICATION OR USE OF ANY MATERIAL SENT IN UNDER A FALSE NAME.

YOU ARE WELCOME TO PHONE ME IF YOU HAVE PROBLEMS.
PHONE 761-451 AFTER 6.00 PM.

—W— RESISTOR FINDER --W--

- by Ken Clarke.

```
0 CLS: REM "RESISTOR FINDER BY KEN CLARKE  2 SEPT 1984
1 PRINT@64,"                                  :7s,A,2Y,I,Y,I,Y,I,9Y,S
2 PRINT" ------|  | | |  ZL1ACS |------       :I,2s,I,s,I,s,I,10s,U
3 PRINT"                                      :7s,D,2T,I,T,I,T,I,9T,F
4 FORP=0TO15:FORQ=1TO8:COLORQ:NEXT:SOUNDP,1:NEXT
5 DIMA$(9),B$(9),C$(9)                        :copy for lines 70-90
10 FORI=0TO9
11 READA$(I):B$(I)=A$(I):C$(I)=A$(I):NEXT
15 REM
20 PRINT:PRINT:PRINTTAB(134);"RESISTOR COLOUR CODE":PRINT
30 SOUND28,1:INPUT"        1ST COLOUR";A$
40 SOUND23,1:INPUT"        2ND COLOUR";B$
50 SOUND20,1:INPUT"        3RD COLOUR";C$
60 GOSUB220:SOUND20,1;23,1;28,1
70 PRINT:PRINT"
80 PRINT" ------|  | | |          |------
90 PRINT"
100 D=((A*10)+B)*C
110 IFD<1000,150
120 IFD<1000000,160
130 PRINT:PRINTTAB(6);" VALUE="D/1000000"M.OHMS
140 GOTO190
150 PRINT:PRINTTAB(6);" VALUE = "D"OHMS"
155 GOTO190
160 PRINT:PRINTTAB(6);" VALUE = "D/1000"K.OHMS"
190 PRINT:PRINT"     PRESS <SPACE> TO REPEAT"
195 IFINKEY$<>" ",195ELSE15
220 FORI=0TO9
230 IFA$=A$(I),A=I
240 IFB$=B$(I),B=I
250 IFC$=C$(I),C=10^I
260 NEXT
270 IFA$="BLACK",A=1
280 IFC$="GOLD",C=.1
290 IFC$="SILVER",C=.01
300 RETURN
310 DATABLACK,BROWN,RED,ORANGE,YELLOW,GREEN,BLUE,VIOLET
320 DATAGREY,WHITE
```

NOTES on Graphics

The codes after the graphics give the number of spaces or graphics and the keys, where to find the graphics.

++++++++++++++++++++++++++++++++++++

Having trouble memorising the Colour Code for resistors?

Solve the problem with the 'Resistor Finder'.

This program could easily be converted to find the values of polyester and tantalum capacitors which use the same colour bands, but in picofarads instead of ohms.

MODE(1) There are other ways of handling designs in MODE(1). One way, often used for complicated designs, is to put the **X** and **Y** co-ordinates into lines of **DATA** and using **READ**. Here is an example of such a program.

```
0 REM MICRO MAGIC
1 REM MAP OF AUSTRALIA
10 MODE(1):COLOR7,0:Y=12
11 B=2:GOSUB100:B=4:GOSUB100:B=11:GOSUB100:B=14:GOSUB100
12 B=11:GOSUB100:B=11:GOSUB100:B=11:GOSUB100:B=7:GOSUB100
13 B=12:GOSUB100:B=12:GOSUB100:B=9:GOSUB100:B=10:GOSUB100
14 B=6:GOSUB100:B=14:GOSUB100:B=5:GOSUB100:B=4:GOSUB100
15 B=5:GOSUB100:B=4:GOSUB100:B=3:GOSUB100:B=2:GOSUB100
16 B=3:GOSUB100:B=2:GOSUB100:B=4:GOSUB100:B=3:GOSUB100
17 B=6:GOSUB100:B=5:GOSUB100:B=5:GOSUB100:B=5:GOSUB100
18 B=5:GOSUB100:B=4:GOSUB100:B=20:GOSUB100:B=14:GOSUB100
19 B=17:GOSUB100:B=14:GOSUB100:B=12:GOSUB100:B=16:GOSUB100
20 B=9:GOSUB100:B=2:GOSUB100:B=2:GOSUB100:B=9:GOSUB100
21 B=5:GOSUB100:B=2:GOSUB100:B=2:GOSUB100:B=3:GOSUB100
22 B=5:GOSUB100:B=3:GOSUB100
23 GOTO50
24 REM
25 REM
30 DATA86,87,57,58,86,88,55,56,57,60,61,62,69,71,85,86,88,60,61
31 DATA62,63,64,65,66,67,68,69,70,85,86,89,56,57,58,59,60,68,69
32 DATA70,85,89,90,49,50,51,52,53,56,57,68,85,89,90,91,47,48,49
33 DATA53,54,55,67,70,85,90,91,46,47,67,84,85,90,91,45,46,67,68
34 DATA69,70,71,83,84,91,92,93,41,42,43,44,45,71,72,73,74,75,83
35 DATA93,39,40,41,75,76,77,82,83,93,37,38,39,77,78,79,80,81,82
36 DATA93,33,34,35,36,37,93,22,24,25,26,27,28,29,30,31,32,33,93
37 DATA94,95,21,22,23,24,95,21,22,95,96,20,21,96,97,98,19,20,98
38 DATA99,19,99,19,99,19,99,100,19,100,17,19,20,100,18,20,100
39 DATA19,20,21,98,99,100,20,21,97,98,99,21,22,96,97,98,22,23
40 DATA94,95,96,23,24,25,93,94,24,25,26,93,26,27,47,48,49,50,51
41 DATA52,53,54,55,56,57,58,59,60,61,62,63,93,27,43,44,45,46,47
42 DATA63,64,65,66,73,74,75,93,26,27,37,38,39,40,41,42,43,66,67
43 DATA68,72,73,75,93,94,27,28,35,36,37,68,69,70,71,72,75,91,92
44 DATA93,28,29,33,34,35,70,71,74,75,89,90,91,29,30,31,32,33,75
45 DATA76,77,78,79,80,84,85,87,88,89,69,70,76,80,81,82,83,84,86
46 DATA84,88,76,78,78,79,80,81,85,86,87,88,89,78,82,83,84,88,79
47 DATA88,80,88,80,81,87,81,84,85,86,87,82,83,84
48 REM
49 REM
50 COLOR6,0
51 Y=19:FORI=1TO23:SET(54,Y):Y=Y+1:NEXT
52 X=55:FORI=1TO26:SET(X,31):X=X+1:NEXT
53 Y=23:FORI=1TO8:SET(76,Y):Y=Y+1:NEXT
54 Y=31:FORI=1TO16:SET(80,Y):Y=Y+1:NEXT
55 X=81:FORI=1TO12:SET(X,34):X=X+1:NEXT
56 X=92:FORI=1TO5:SET(X,33):X=X+1:NEXT
57 X=96:FORI=1TO3:SET(X,34):X=X+1:NEXT
58 SET(98,33):SET(99,33)
59 SET(80,41):SET(81,41):SET(82,41):SET(89,43):SET(90,43)
60 SET(91,43):SET(92,43)
61 X=83:FORI=1TO7:SET(X,42):X=X+1:NEXT
62 GOTO62
100 FORA=1TOB:READX:SET(X,Y):NEXT:Y=Y+1:RETURN
```

and then there is POKE...

A Study of the MODE(1) VIDEO DISPLAY WORKSHEET (Screen Grid) will reveal an important fact. Looking at the corners, you'll see the Memory Locations, which are given, in HEXIDECIMAL and DECIMAL (in brackets), refer to 4 pixels, not one.
The Memory Addresses for POKEs are the Decimal locations.
If we were to POKE 28672 with any number between Ø and 255 then the four top left hand pixels would be changed. Some numbers POKEd into 28672 would " turn on " some of the pixels in a color leaving the rest " turned off". Other numbers would turn on pixels in different colors. There are 256 permutations.
Here is a program for drawing borders as in **MICROMAZE...**

```
Ø PROGRAM FOR MODE(1) BORDERS.
1ØØ MODE(1):COLOR3,1
11Ø S=28672:GOSUB2ØØ:S=S+32:GOSUB 2ØØ
12Ø S=3Ø624:GOSUB2ØØ:S=S+32:GOSUB 2ØØ
13Ø S=28736:GOSUB21Ø:S=S+31:GOSUB 21Ø
14Ø GOTO14Ø
2ØØ FORI=STOS+31:POKEI,17Ø:NEXT: RETURN
21Ø FORI=STOS+192ØSTEP32:POKEI,17Ø: NEXT:RETURN
```

Here is a program that you may like to experiment with. It prints a number on the screen and then changes to MODE(1) and shows you the effect of POKING that number into a MODE(1) display. By pressing P the next number is printed and then displayed.

```
1ØØ CLS:CLEAR:FORP=ØTO255:PRINT@27Ø ,P:FORT=1TO5ØØ:NEXT
11Ø MODE(1):S=29612
12Ø FORI=1TO8:FORJ=STOS+8:POKEJ,P: NEXT:S=S+32:NEXT
13Ø Z$=INKEY$:Z$=""
14Ø Z$=INKEY$:IFZ$=""THEN14Ø
15Ø IF Z$="P" THENNEXT
16Ø GOTO14Ø
```

This next program allows you to INPUT Ø to 255 which will be POKEd into MODE(1) screen locations.

```
1ØØ CLS:CLEAR:COLOR,1
11Ø PRINT@235,"";:INPUTP:MODE(1):S= 29612
12Ø FORI=1TO8:FORJ=STOS+8:POKEJ,P: NEXT:S=S+32:NEXT
13Ø Z$=INKEY$:Z$=""
14Ø Z$=INKEY$:IFZ$=""THEN14Ø
15Ø IFZ$="P"THEN11Ø
16Ø GOTO14Ø
```

Here is a TABLE derived from the above programs...

POKE	COLOR,Ø	COLOR,1	
,Ø	GREEN	BUFF	(BACKGROUND COLORS)
,85	YELLOW	CYAN	
,17Ø	BLUE	MAGENTA	
,255	RED	ORANGE	

...and the advantage of POKEs over SETs is SPEED !

SOFTWARE REVIEW AND PREVIEW

Review of TEAPOT TED
Written by Brenda Hill.

The editor of this magazine gave me a frantic phone call early in July, Hoping that I had or could write a software review for him as he was short of items for the next issue of the magazine.
So I hope he likes this short review on Teapot Ted.

The editor gave me a disk containing programs from the newsletters and on the end of this disk was a lovely childrens song.
As soon as I heard it, all my child hood years came flowing back, memories of my mother sitting there with 6 kids teaching them childrens songs.
Like most of us, we grew up on "I'm a little teapot".
(All except one, not naming anyone (P.J)).

The computer version of TEAPOT TED was written by R. Winter. And is a credit to him or her.
The music and graphics of this childs song are well done.
It's a catchy tune for kids to listen to, at any age.

Entered and submitted from the Bumper book of programs by Ken Clarke.

VZ 200-300 PCB MINICAD DESIGNER

This program is primarily designed for use with the PP-40 or TP-40 PRINTER/PLOTTER, although other X-Y PLOTTERS with the same command structure may be used. The software is only compatable with VZ Disc Operating Systems either VZ200 or VZ300 with 16K EXPANSION or MEMORY EXTENDING ABOVE B800 hex.

The program will cater for the hobbyist in that it will produce a neat and accurate PCB layout for small PCB's containing up to 10 IC's and caters for various transistor packages and other small components.

When run the program will firstly check for a printer 'on line', if not it will pause with a prompt until the printer is available. After the printer check is complete, the main menu will appear giving 9 options and the STATUS of the disk in the specified DISK DRIVE.

As with any of the options chosen from the MAIN MENU that involve creation of data, you are given the option to insert a new disk and elect the Disk Drive to which the generated data is sent. The format of the software directs all generated data to the Disk in a Sequential File which means the limit of data created is not that of FREE MEMORY in the computer but 78K of FREE DISK or multiples of this if you elect to spread the data over more than one disk. With this in mind a NEW DISK is recommended. BE SURE THAT THE NEW DISK HAS BEEN FOMATTED WITH THE INIT COMMAND BEFORE THE PROGRAM IS RUN.

When you are prompted for DISK FILE TITLE it should be remembered that only 8 characters are allowed and only the first 8 are used by the program if any more are entered.

TEAPOT TED

```
1 POKE30771,1:'"[illegible]
5 GOSUB100:GOSUB108:GOSUB115
6 FORT=1TO500:NEXTT
10 PRINT@43,"I'M A LITTLE TEA-POT"
11 SOUND16,2;18,2;20,2;21,2;23,4;28,4
15 PRINT@35,"          "
16 PRINT@67,"          "
17 PRINT@99,"          "
18 GOSUB102:GOSUB108
19 PRINT@76,"SHORT... "
21 SOUND25,4
25 PRINT@288,"  [illegible] "
26 PRINT@320,"  [illegible] "
27 PRINT@352," [illegible]"
29 PRINT@85,"AND STOUT."
30 SOUND28,4;23,6
35 PRINT@256," [illegible] "
36 PRINT@288," [illegible] "
37 PRINT@320," [illegible] "
39 PRINT@141,"HERE IS MY HANDLE,"
40 SOUND21,4;21,2;21,2;20,4;20,4
41 PRINT@256," [illegible]"
42 PRINT@288," [illegible]   "
43 PRINT@320," [illegible]   "
44 PRINT@352," [illegible]   "
45 PRINT@224,"     [illegible]  "
46 PRINT@192,"    [illegible]  [illegible]"
50 PRINT@174,"HERE IS MY SPOUT."
54 SOUND18,4;18,2;18,2;16,6
55 PRINT@43,"   WHEN I GET ALL   "
56 PRINT@76,"   STEAMED UP      "
58 PRINT@141,"                  "
59 PRINT@172,"                    "
60 SOUND16,2;18,2;20,2;21,2;23,4;28,4
61 PRINT@96,"    \!!/   "
62 FORT=1TO20:NEXTT
63 PRINT@64,"   0 00 0  "
64 FORT=1TO20:NEXTT
65 PRINT@32,"  \ \!!/ / "
66 FORT=1TO20:NEXTT
67 PRINT@0," 0 0 00 0 0"
69 PRINT@142,"HEAR ME SHOUT... "
70 SOUND25,4;28,4;23,6
71 PRINT@0,"                 "
72 PRINT@32,"                              "
75 PRINT@64,"  TIP ME OVER....            "
76 PRINT@96,"              "
78 PRINT@142,"                 "
79 SOUND28,6;25,2;23,4;21,4
80 PRINT@192,"    [illegible]         "
81 PRINT@224,"     [illegible]      [illegible] "
82 PRINT@256," [illegible] [illegible]  [illegible]"
83 PRINT@288," [illegible] [illegible]    "
86 PRINT@384,"    [illegible] "
87 PRINT@416,"   [illegible]  [illegible] "
88 PRINT@448,"       [illegible]"
89 PRINT@81,"POUR ME OUT!":SOUND20,4;18,4;16,6
90 PRINT@432," [illegible]":PRINT@464," [illegible] "
91 FORT=1TO200:NEXTT
92 PRINT@304,"V":SOUND16,1:PRINT@336,"V":SOUND15,1:PRINT@369,"V"
```

```
93  SOUND14,1:PRINT@401,"V":SOUND13,1:FORT=1TO20:NEXTT:SOUND12,8
95  PRINT@480," ":PRINT"     HIT 'Y' TO RUN AGAIN";
96  A$=INKEY$:A$=INKEY$:IFA$="Y"GOTO5
97  IFA$=""GOTO96
99  IFA$<>"Y"GOTO96
100 CLS
101 PRINT
102 PRINT"     ▄▄▄     "
103 PRINT"   ▄▄▄▄▄▄▄   "
104 PRINT"    ▄▄▄▄▄    "
105 PRINT"      ▄      "
106 PRINT"  ▄▄▄▄▄▄▄▄▄  "
107 RETURN
108 PRINT"  ▄ ▄▄▄▄ ▄  "
109 PRINT"  ▄ ▄▄▄▄ ▄  "
110 PRINT" ▄▄ ▄▄▄▄▄ ▄▄ "
112 PRINT"    ▄▄▄▄▄  "
113 PRINT"    ▄   ▄  "
114 RETURN
115 PRINT"    ▄   ▄  "
116 PRINT"    ▄   ▄  "
117 PRINT"    ▄   ▄  "
118 PRINT"   ▄▄  ▄▄  "
120 RETURN
```

RESTORE BY DAVE MITCHELL - - -

SO YOU PRESSED NEW INSTEAD OF RUN

DON'T WORRY :- When you use the NEW command the VZ POKES TWO ZERO BYTES into the start of the basic program to replace these bytes is very easy What they tell the computer is the address of the next line of the basic program.

TRY THIS :- Type in line 10 then return
Type in line 20 then return

```
10 CLS
20 PRINT
```

WITH NO LINE NUMBER TYPE & press return

```
FOR I = 31465 to 31476 : PRINT I ; PEEK( i ): NEXT
```

LISTED ON THE VIDEO SHOULD BE

```
31465  239   ; ADDRESS OF THE NEXT LINE (LSB)
31466  122   ; ADDRESS OF THE NEXT LINE (MSB)
31467   10   ; LINE NUMBER (LSB)
31468    0   ; LINE NUMBER (MSB)
31469  132   ; TOKEN FOR CLS
31470    0   ; END OF LINE
31471  245   ; ADDRESS OF NEXT LINE (LSB)
31472  122   ; ADDRESS OF NEXT LINE (MSB)
31473   20   ; LINE NUMBER (LSB)
31474    0   ; LINE NUMBER (MSB)
31475  178   ; TOKEN FOR PRINT
31476    0   ; END OF LINE
```

31465 & 31466 contains the address of the next line. WITH simple maths it is easy to find this address :-

239 + 256 * 122 = 31471

CHECK 31471 YOU WILL SEE IT IS THE ADDRESS OF THE NEXT Line.

List the program you typed. Now press NEW & return.
List again, WHAT DID YOU GET ? WAS IT NOTHING ?

TYPE :- PRINT PEEK(31465) ; PEEK(31466) & press return.
IT SHOULD BE --- 0 0

TYPE :- POKE 31465,239 : POKE 31466,122 : LIST & press return

You have just got the program back from the UNKNOWN !

Here is an example in basic of what the machine code does.

```
10 A = 31467
20 B = PEEK( A )
30 IF B <> 0 THEN A=A+1 : GOTO 20
40 IF B = 0 THEN A=A+1
50 C=A-(256*INT(A/256)
60 D=INT(A/256)
70 POKE 31465,C
80 POKE 31466,D
```

In line 10 'A' equals the normal start of the basic program plus two.
Line 20 'B' looks at what is at location 'A'
Line 30 checks if 'B' is bigger or smaller than zero and adds one to 'A' then goes to line 20
Line 40 checks if 'B' equals zero then 'A' is added with one.(the zero inicates the end of the basic line.)
Line 50 works out the LEAST SIGNIFICIANT BYTE (LSB).
Line 60 works out the MOST SIGNIFICIANT BYTE.
Line 70 POKES 31465 with the LSB.
Line 80 POKES 31466 with the MSB.

The above program is only an example to help you understand the assembler code later.

FOR THOSE WITHOUT AN ASSEMBLER PROGRAM.

```
10 FOR I = -20480 TO -20391
20 READ A:POKE I,A: NEXT
30DATA205,201,1,33,39,176,205,167,40,17,236,122,19,26,254,0,32
40DATA250,19,235,34,233,122,33,233,122,34,164,120,205,248,26,35
50DATA34,249,120,195,25,26,13,32,32,32,32,32,32,32,32,32,32,32
60DATA82,69,83,84,79,82,69,13,13,32,32,32,32,32,66,89,32,68,46
70DATA77,73,84,67,72,69,76,76,32,32,86,75,52,75,68,65,0,0,0,0
80 CLS:PRINT"SAVE TO DISK OR TAPE   (D/T)"
90 A1$=INKEY$:A$=INKEY$:IFA$<>"D"ANDA$<>"T"THEN90
100 SOUND30,1:IFA$="T"THEN180
110 IF PEEK( 16384 ) = 170,140
120 PRINT"NO DISK DRIVE YOU NAUGHTY PERSON"
130 GOTO 180
140 PRINT"INSERT DISK,CLOSE DOOR & PRESS  RETURN"
150 IFINKEY$<>CHR$(13),150
155 SOUND30,1
160 BSAVE"RESTORE",B000,B059
170 END
180 FOR I = -24576 TO -24526
190 READ A: POKE I,A: NEXT
200 PRINT"INSERT CASSETTE,PRESS PLAY &    RECORD THEN RETURN"
```

```
210 IF INKEY$ <> CHR$(13),210
220 SOUND30,1
230 POKE 30862,0:POKE 30863,160:X=USR(0)
240 END
250 DATA33,0,176,34,164,120,33,89,176,34,249,120
260 DATA33,38,160,14,241,243,205,172,52,251,33,233,122
270 DATA34,164,120,205,248,26,35,34,249,120
280 DATA195,25,26,34,82,69,83,84,79,82,69,34,0,0,0,0
```

Lines 10 to 70 contains the data for restore.
Lines 80 to 100 checks for saving to disk or tape.
Line 110 checks for a disk drive.
Lines 120 to 130 prints an error then goes to tape save.
Lines 140 to 170 saves to disk.
Lines 180 to 190 sets up data for AUTO-RUN tape save.
Lines 200 to 210 prints message & waits for the return key to be pressed.
Line 230 jumps to the machine code for AUTO-RUN TAPE SAVE.
Lines 250 to 280 data for auto-run tape save.

FOR MULTIPLE COPIES :- DISK TYPE GOTO 140
TAPE TYPE GOTO 200.

THE ASSEMBLER CODE.

```
     CALL 01C9H    ;clear screen.
     LD HL,MES     ;load HL with the message address.
     CALL 28A7H    ;print the message.
     LD DE,7AECH   ;load de with 7AEC hex.
SCAN INC DE        ;add one to de.
     LD A,(DE)     ;load 'A' with what is at the address of DE.
     CP 00H        ;test - is it a zero.
     JR NZ,SCAN    ;NO - go back to scan & do it again.
                   ;YES - continue.
     INC DE        ;add one to DE.
     EX DE,HL      ;exchange regesters.
     LD (7AE9H),HL;load the contents of HL into 7AE9 HEX.
     LD HL,7AE9H   ;load HL with 7AE9 HEX.
     LD (78A4H),HL;load start of program with HL.
     CALL 1AF8H    ;reset program statement table.
     INC HL        ;add one to HL.
     LD (78F9H),HL;load end of program with HL.
     JP 1A19H      ;JUMP TO BASIC.
MES  EQU $         ;start of message.
     DEFB 0DH      ;carriage return.
* [11 spaces] RESTORE*
     DEFB 0DH      ;carriage return.
     DEFB 0DH      ;carriage return.
* [5 spaces] BY D.MITCHELL*
* [2 spaces] VK4KDA*
     NOP           ;end of message.
```

ORGIN 0B00 HEX.
START OF PROGRAM B000 HEX.
END OF PROGRAM B059 HEX.

NOTE :- PROGRAM WILL RUN IN A STANDARD VZ300 OR VZ200 + 16K.

Each article in this series is a gentle introduction to some topic in the computing field. It is written for the beginner, so may appear very simple to the rest of you. If you find it too easy, and so not worth reading, congratulations — you are a beginner no more. Each issue will deal with a different topic, of general interest. Occasionally, material may seem to repeat what has already appeared in "Bits and Bytes". But remember, new readers are coming along all the time.

Peripherals

By Gordon Findlay

Mention a computer and most people immediately think of a processor — a piece of hardware which can do the actual number work, make decisions and so on. But that isn't all there is to a computer system, by any means.

A peripheral is any piece of hardware which forms part of a computer system but which isn't part of the main unit. The processor and the main memory (that mysterious ROM and RAM) are part of the main unit. So probably, is a language interpreter for say BASIC, and a few interfaces. It is these interfaces which provide access to all the peripherals you want — or can afford!

What is an interface? I'll avoid detail just now, and devote a future article to them, but for now an interface is a piece of electronics between two other pieces of electronics (such as a computer and a printer) which enables them to talk to each other. It must convert from the electrical terms one deals with to the terms the other can cope with. That is sometimes dead easy, and sometimes it isn't. Most computers have a range of interfaces either built in or available to add on.

There are a lot of peripherals on the market and it isn't always straightforward to decide which are wanted or needed. Let's look at them in turn.

The most elementary, and most essential peripheral is a monitor. Naturally, you need a screen to see what is going on! The screen might be the family TV set, another TV, or a special unit designed solely for use with a computer. These special units are what is properly known as a monitor.

A monitor proper is much better than a TV set. Why? The monitor will give a better image — clearer letters, usually decaying slightly more slowly. A TV set just isn't clear enough for more than 40 letters across the screen at a maximum. Better monitors can cope with up to 132 characters easily. The measurement to be concerned with is the bandwidth — the higher its value the better.

Amber screen on its way

Of course, the TV set may be coloured — as may a monitor at a price. Monochrome monitors ("one colour") used to be all black and white, but nowadays are more likely to have green screens. Green is regarded as much easier to tolerate for a long time than black and white. Amber screens have recently appeared, and I predict the amber screen will replace the green one in a very short time. Amber is the standard colour for public service use in many European countries, and is meant to be even better than green for prolonged use.

Blue is said to be even better, and red screens have been advertised. Clearly, differences between the colours are subjective to some extent. My personal preference is for a blue screen, but that is based on only a short trial, and could be misleading.

The next major peripheral is some form of off-line or permanent storage, such as a tape recorder or disk drive. A lot of ink has been spilt over the relative advantages and disadvantages of the various types. Of course, tape is much cheaper — that's the only reason for using it. Obviously, disks are faster and more versatile as well as expensive. There are things such as storing large volumes of complex data which just aren't feasible from tape, but which are very elementary operations for a disk system.

There are lots of different types of disk drives — far too many in some opinions! One firm in Christchurch has a device for changing between disk formats — more than 115 of them!

For all business purposes, a printer is essential and it is very desirable at home too. There are literally hundreds of different printers available — far too many to even summarise here. *Bits and Bytes* has covered the field comprehensively in earlier issues. This is the point at which you really need to come to grips with the different forms of interface and make sure the computer and printer are compatible.

To communicate with another computer, you need a modem. This device allows you to connect your computer to a telephone, and so communicate with others via the telephone system. This opens up all sorts of possibilities. Not only can computers share information, a home computer can access information "banks" or databases, and bulletin boards. This is reputedly the fastest growing area in computing in the USA, and it looks poised for a major expansion here too. Expect to see major developments in the next 12 to 18 months.

Tired of typing? A mouse might be just for you. It isn't something which eats holes in your computer, but a small box running on a ball. Move the mouse and the cursor on the screen moves. The mouse can be used to perform all sorts of operations just by pointing an arrow on the screen with it, and clocking a button on the top. Naturally, you need software which understands mouses (mice?).

Plotters draw pictures on paper. Joysticks and track-balls give control over games. Real time clocks keep track of the time, and even act as an alarm and on it goes. Oh for a money tree!